Europa Clipper: MBSE Proving Ground

Todd Bayer

Chief Engineer, Flight Systems Engineering, Integration and Test Section

with

John Day, Emma Dodd, Laura Jones-Wilson, Andres Rivera, Narek Shougarian, Sara Susca, David Wagner

For GSFC Systems Engineering Seminar

28 July 2021

Jet Propulsion Laboratory
California Institute of Technology

Topics

- Introduction
- MBSE Applications on Europa Clipper - Summary
- More details on each:
  - MEL & PEL
  - Power/Energy Modeling and Simulation
  - Architecture and Requirements Development
  - Science Traceability
  - Electrical Interfaces and Harness Specification
- Conclusion

The Future of Engineering is Model-Based

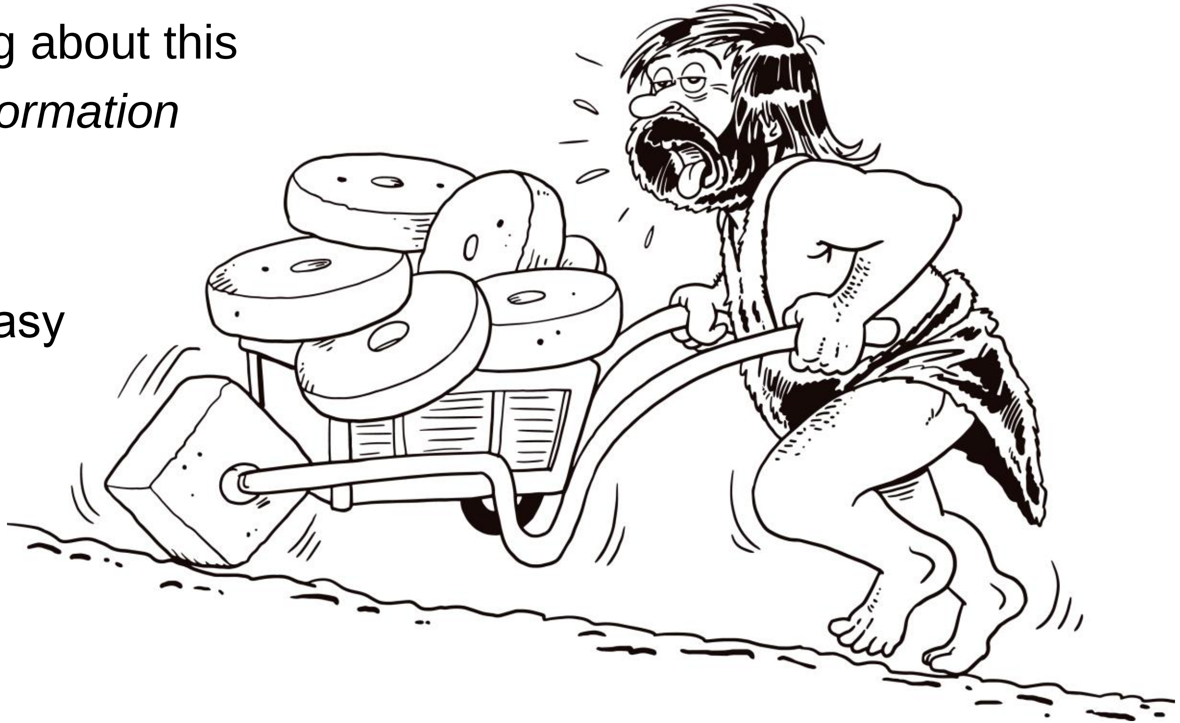
Model-Based *Systems Engineering*

is key to bringing about this

Digital Transformation

But that doesn't mean it's easy

or obvious how!



Introduction

Since ~2010

- **JPL's Integrated Model Centric Engineering Initiative (IMCE) has led MBSE infusion at JPL**
- **IMCE has had active collaborations in several areas with the Europa Project**

Many have been successful

…Others have not

But all have been valuable learning experiences

We'll discuss examples and lessons from each

MBSE Applications on Europa Clipper

Application	Description	In use
Mass Equipment List (MEL)	• SysML/Magid Draw capture and rollup of component mass • Web-based reporting via OpenMBEE	2011-present
Power Equipment List (PEL)	• Add Power states/demands to MEL • Provide static description to time-based mission simulation • Web-based reporting via OpenMBEE	2012-present
Power/Energy Simulation	• Multiple tools chained to provide repeatable power demand, energy production,& battery state of charge profiles	2013-present
Architecture & Requirements	• SysML/MD and View Editor (OpenMBEE) -based architecture and requirement development tools	Partial capability 2014; Retired unfinished 2019
Science Traceability and Alignment Framework (STAF)	• Framework for tracing science measurement requirements to project, spacecraft and science instrument requirements • Excel-based	2016 - present
Electrical Systems Engineering	• Eclipse/EMF-based authoring tool, git for CM, Leveraging open-source standards (OWL2-DL, SPARQL)	2019-present

MEL and PEL: Mass and Power lists

Table 1.2. Orbiter Flight System WBS-Based MEL

	Number of Units	Launch Mass		
		Mass State		
		Mass Current Best Estimate	Mass Contingency	Mass CBE_+_Contingency
00 Orbiter Flight System	1	1344.89	1.32	1772.64
05 Orbiter Payload	1	20.21	1.50	30.32
LA	1	10.15	1.50	15.23
LA	1	7.95	1.50	11.93
LA Sensor	1	3.25	1.50	4.88
LA Sensor Shielding	1	4.70	1.50	7.05
LA Card	1	0.90	1.50	1.35
LA PCU Card	1	1.30	1.50	1.95
LP	1	2.74	1.50	4.11
LP Card-1	1	0.90	1.50	1.35
LP Card-2	1	0.90	1.50	1.35
LP-1	1	0.47	1.50	0.70
LP Sensor	1	0.47	1.50	0.70
LP Sensor Shielding	1	0.00	1.50	0.00
LP-2	1	0.47	1.50	0.70
LP Sensor	1	0.47	1.50	0.70
LP Sensor Shielding	1	0.00	1.50	0.00
MAG	1	3.32	1.50	4.98
MAG	1	2.42	1.50	3.63
MAG Sensor	1	2.42	1.50	3.63
MAG Sensor Shielding	1	0.00	1.50	0.00
MAG Card	1	0.90	1.50	1.35
Mapping Camera	1	4.00	1.50	6.00
Mapping Camera	1	3.10	1.50	4.65
Sensor	1	1.60	1.50	2.40
Sensor Shielding	1	1.50	1.50	2.25
Mapping Camera Card	1	0.90	1.50	1.35
06 Orbiter Spacecraft	1	1324.68	1.32	1742.32

Table 1.5. Orbiter Flight System WBS-Based PEL

	Number of Units	Power Timeline					
		Power Off		Power On		Power Standby	
		Power State Prototype		Power State Prototype		Power State Prototype	
		Power Contingency	Power Current Best Estimate	Power Contingency	Power Current Best Estimate	Power Contingency	Power Current Best Estimate
00 Orbiter Flight System	1	1.30	0.00	1.30	0.00	1.30	0.00
05 Orbiter Payload	1	n/a	n/a	n/a	n/a	n/a	n/a
LA	1	n/a	n/a	n/a	n/a	n/a	n/a
LA	1	1.30	0.00	1.30	15	1.30	0.00
LA Sensor	1	1.30	0.00	1.30	0.00	1.30	0.00
LA Sensor Shielding	1	n/a	n/a	n/a	n/a	n/a	n/a
LA Card	1	1.30	0.00	1.30	0.00	1.30	0.00
LA PCU Card	1	1.30	0.00	1.30	0.00	1.30	0.00
LP	1	n/a	n/a	n/a	n/a	n/a	n/a
LP Card-1	1	1.30	0.00	1.30	0.00	1.30	0.00
LP Card-2	1	1.30	0.00	1.30	0.00	1.30	0.00
LP-1	1	1.30	0.00	1.30	1.15	1.30	0.00
LP Sensor	1	1.30	0.00	1.30	0.00	1.30	0.00
LP Sensor Shielding	1	n/a	n/a	n/a	n/a	n/a	n/a
LP-2	1	1.30	0.00	1.30	1.15	1.30	0.00
LP Sensor	1	1.30	0.00	1.30	0.00	1.30	0.00
LP Sensor Shielding	1	n/a	n/a	n/a	n/a	n/a	n/a
MAG	1	n/a	n/a	n/a	n/a	n/a	n/a
MAG	1	1.30	0.00	1.30	4	1.30	0.00
MAG Sensor	1	1.30	0.00	1.30	0.00	1.30	0.00
MAG Sensor Shielding	1	n/a	n/a	n/a	n/a	n/a	n/a
MAG Card	1	1.30	0.00	1.30	0.00	1.30	0.00
Mapping Camera	1	n/a	n/a	n/a	n/a	n/a	n/a
Mapping Camera	1	1.30	0.00	1.30	6.00	1.30	0.00
Sensor	1	1.30	0.00	1.30	0.00	1.30	0.00
Sensor Shielding	1	n/a	n/a	n/a	n/a	n/a	n/a
Mapping Camera Card	1	1.30	0.00	1.30	0.00	1.30	0.00

MBSE Applications on Europa Clipper – with Key Benefits and Lessons

Application	Description	In use	Key Benefits	Key Lessons
Mass Equipment List (MEL)	• SysML/Magid Draw capture and rollup of component mass • Web-based reporting via OpenMBEE	2011-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled early CM	(+) Start small: modest, incremental objectives (+) Produce familiar products with better methods (+) Involve end user continuously (+) Effective project/line collaboration is essential
Power Equipment List (PEL)	• Add Power states/demands to MEL • Provide static description to time-based mission simulation • Web-based reporting via OpenMBEE	2012-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled integrated simulations	(+) Leverage existing tools (+) Include only the necessary data in the model (-) SysML not ideal for tabular UI (-) Integrate early with Electrical SE model
Power/Energy Simulation	• Multiple tools chained to provide repeatable power demand, energy production,& battery state of charge profiles	2013-present		
Architecture & Requirements	• SysML/MD and View Editor (OpenMBEE) -based architecture and requirement development tools	Partial capability 2014; Retired unfinished in 2019		
Science Traceability and Alignment Framework (STAF)	• Framework for tracing science measurement requirements to project, spacecraft and science instrument requirements • Excel-based	2016 - present		
Electrical Systems Engineering	• Eclipse/EMF-based authoring tool, git for CM, Leveraging open-source standards (OWL2-DL, SPARQL)	2019-present		

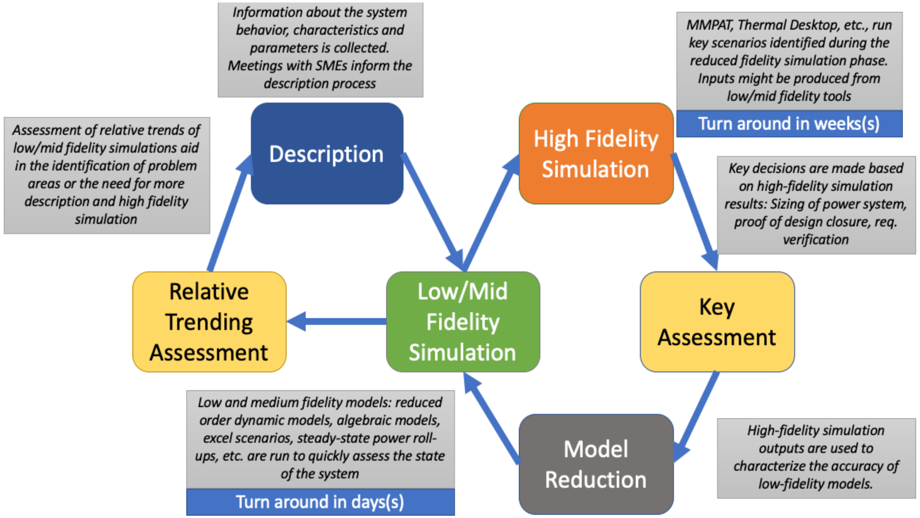


Figure 1. Rapid simulation approach

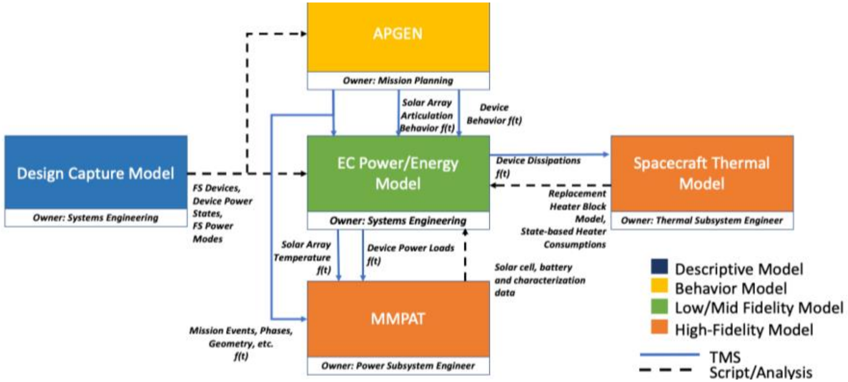
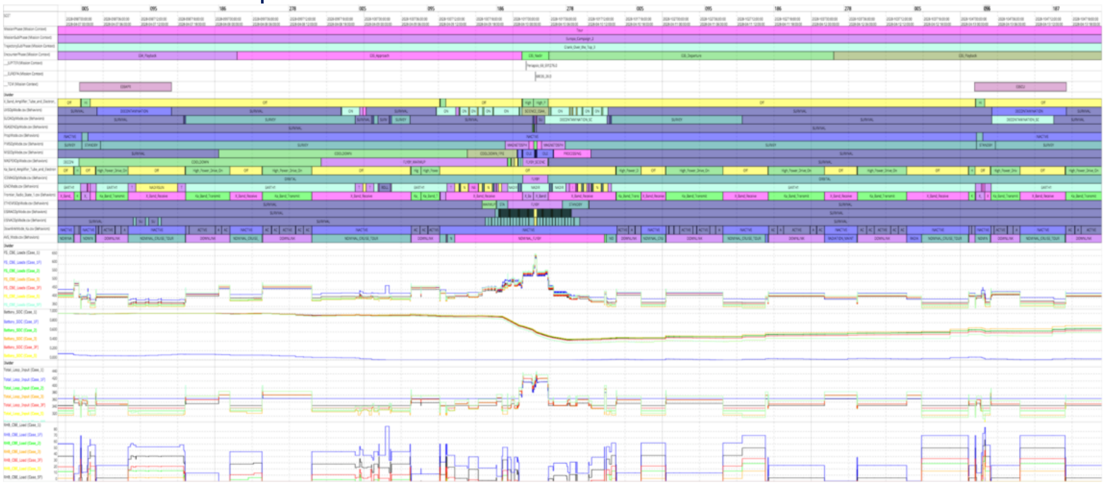


Figure 2. Primary models, interfaces and information flows

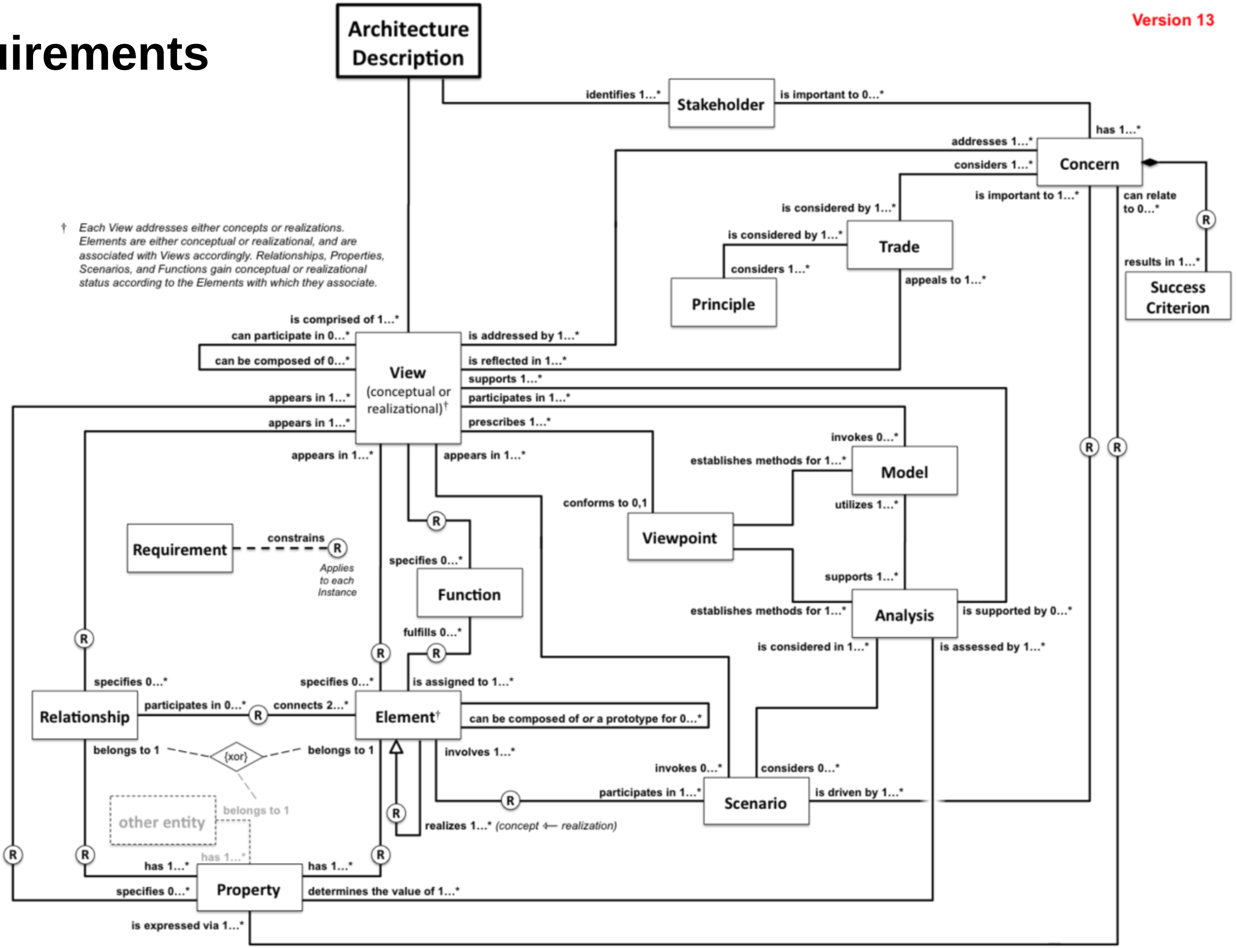
Power/Energy Simulation



MBSE Applications on Europa Clipper – with Key Benefits and Lessons

Application	Description	In use	Key Benefits	Key Lessons
Mass Equipment List (MEL)	• SysML/Magid Draw capture and rollup of component mass • Web-based reporting via OpenMBEE	2011-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled early CM	(+) Start small: modest, incremental objectives (+) Produce familiar products with better methods (+) Involve end user continuously (+) Effective project/line collaboration is essential
Power Equipment List (PEL)	• Add Power states/demands to MEL • Provide static description to time-based mission simulation • Web-based reporting via OpenMBEE	2012-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled integrated simulations	(+) Leverage existing tools (+) Include only the necessary data in the model (-) SysML not ideal for tabular UI (-) Integrate early with Electrical SE model
Power/Energy Simulation	• Multiple tools chained to provide repeatable power demand, energy production,& battery state of charge profiles	2013-present	• Automatic transformation of PEL model resulted in frequent reporting of power and energy resources • Early ability to size system using full mission simulation • Frequent snapshots • Static snapshots in PEL provide validation of time-based simulation	(+) Share time-based profiles with subsystems early on for detailed design (+) Run frequently to detect issues early (+) Validate against high-fidelity subsystem tools (-) Earlier documentation would have helped during leadership change
Architecture & Requirements	• SysML/MD and View Editor (OpenMBEE) -based architecture and requirement development tools	Partial capability 2014; Retired unfinished in 2019		
Science Traceability and Alignment Framework (STAF)	• Framework for tracing science measurement requirements to project, spacecraft and science instrument requirements • Excel-based	2016 - present		
Electrical Systems Engineering	• Eclipse/EMF-based authoring tool, git for CM, Leveraging open-source standards (OWL2-DL, SPARQL)	2019-present		

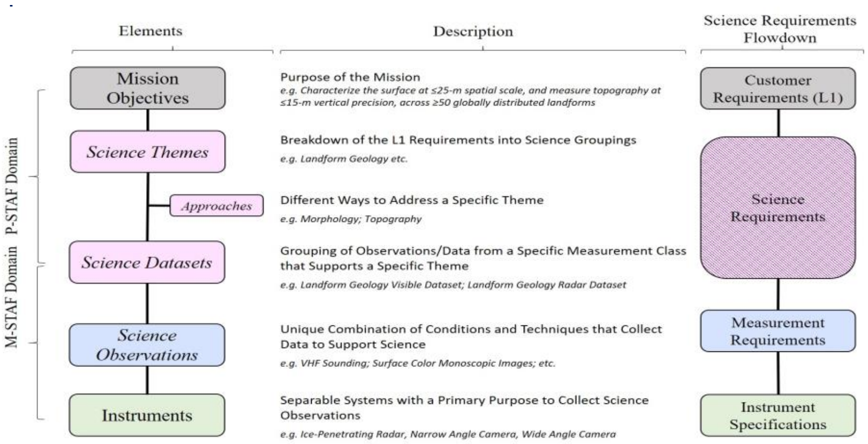
Architecture and Requirements

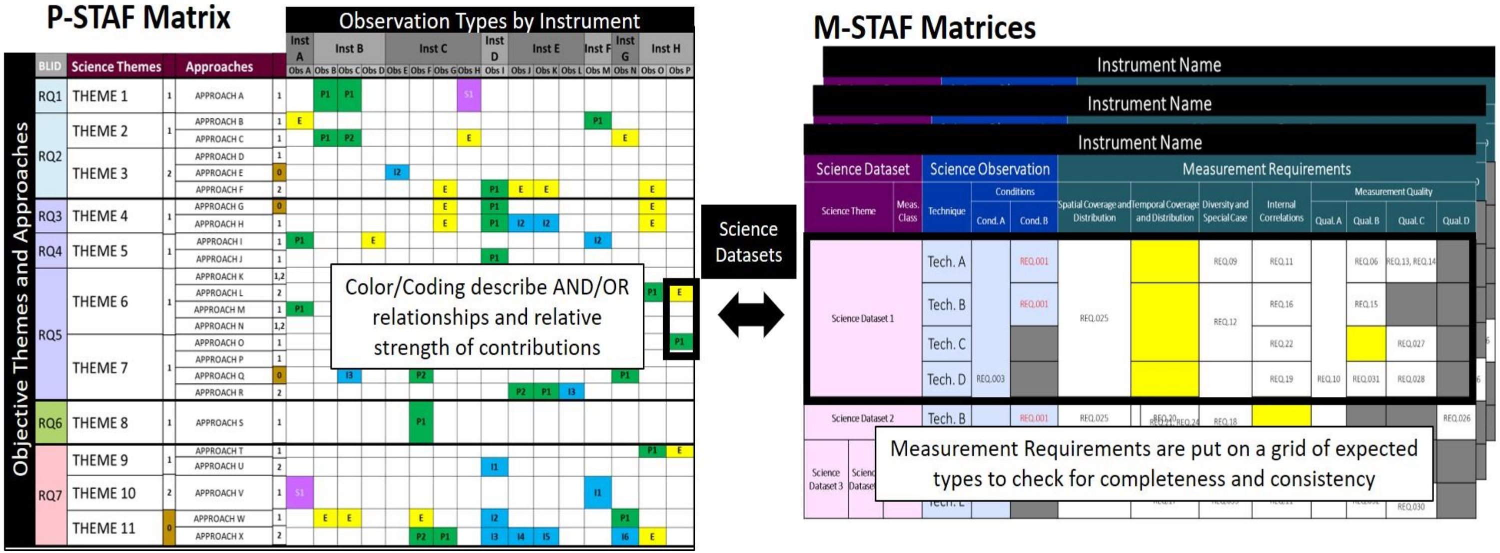


MBSE Applications on Europa Clipper – with Key Benefits and Lessons

Application	Description	In use	Key Benefits	Key Lessons
Mass Equipment List (MEL)	• SysML/Magid Draw capture and rollup of component mass • Web-based reporting via OpenMBEE	2011-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled early CM	(+) Start small: modest, incremental objectives (+) Produce familiar products with better methods (+) Involve end user continuously (+) Effective project/line collaboration is essential
Power Equipment List (PEL)	• Add Power states/demands to MEL • Provide static description to time-based mission simulation • Web-based reporting via OpenMBEE	2012-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled integrated simulations	(+) Leverage existing tools (+) Include only the necessary data in the model (-) SysML not ideal for tabular UI (-) Integrate early with Electrical SE model
Power/Energy Simulation	• Multiple tools chained to provide repeatable power demand, energy production,& battery state of charge profiles	2013-present	• Automatic transformation of PEL model resulted in frequent reporting of power and energy resources • Early ability to size system using full mission simulation • Frequent snapshots • Static snapshots in PEL provide validation of time-based simulation	(+) Share time-based profiles with subsystems early on for detailed design (+) Run frequently to detect issues early (+) Validate against high-fidelity subsystem tools (-) Earlier documentation would have helped during leadership change
Architecture & Requirements	• SysML/MD and View Editor (OpenMBEE) -based architecture and requirement development tools	Partial capability 2014; Retired unfinished in 2019	• Architecture Framework provided a useful mental model adapted to JPL missions • Conceptual approach contributed to more complete requirements • Produced reusable stakeholder descriptions	(-) Ambitious developments need risk management (-) Training the full team to think differently is hard (-) Tools and processes need to support end user (-) Parallel developments impede failure analysis
Science Traceability and Alignment Framework (STAF)	• Framework for tracing science measurement requirements to project, spacecraft and science instrument requirements • Excel-based	2016 - present		
Electrical Systems Engineering	• Eclipse/EMF-based authoring tool, git for CM, Leveraging open-source standards (OWL2-DL, SPARQL)	2019-present		

Science Traceability and Framework

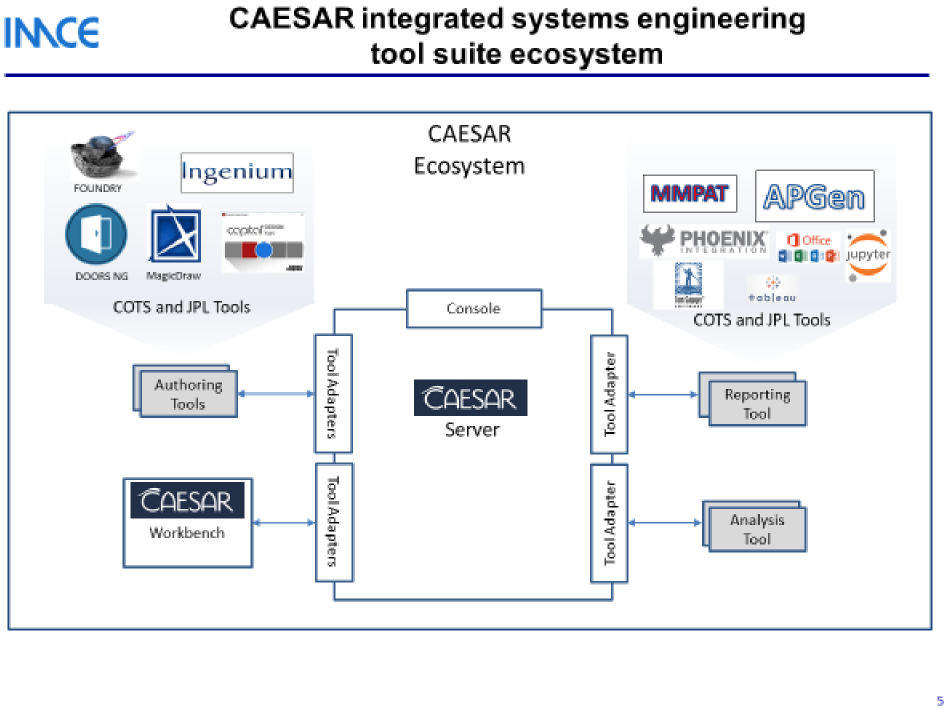




MBSE Applications on Europa Clipper – with Key Benefits and Lessons

Application	Description	In use	Key Benefits	Key Lessons
Mass Equipment List (MEL)	• SysML/Magid Draw capture and rollup of component mass • Web-based reporting via OpenMBEE	2011-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled early CM	(+) Start small: modest, incremental objectives (+) Produce familiar products with better methods (+) Involve end user continuously (+) Effective project/line collaboration is essential
Power Equipment List (PEL)	• Add Power states/demands to MEL • Provide static description to time-based mission simulation • Web-based reporting via OpenMBEE	2012-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled integrated simulations	(+) Leverage existing tools (+) Include only the necessary data in the model (-) SysML not ideal for tabular UI (-) Integrate early with Electrical SE model
Power/Energy Simulation	• Multiple tools chained to provide repeatable power demand, energy production,& battery state of charge profiles	2013-present	• Automatic transformation of PEL model resulted in frequent reporting of power and energy resources • Early ability to size system using full mission simulation • Frequent snapshots • Static snapshots in PEL provide validation of time-based simulation	(+) Share time-based profiles with subsystems early on for detailed design (+) Run frequently to detect issues early (+) Validate against high-fidelity subsystem tools (-) Earlier documentation would have helped during leadership change
Architecture & Requirements	• SysML/MD and View Editor (OpenMBEE) -based architecture and requirement development tools	Partial capability 2014; Retired unfinished in 2019	• Architecture Framework provided a useful mental model adapted to JPL missions • Conceptual approach contributed to more complete requirements • Produced reusable stakeholder descriptions	(-) Ambitious developments need risk management (-) Training the full team to think differently is hard (-) Tools and processes need to support end user (-) Parallel developments impede failure analysis …more on slides 10-11
Science Traceability and Alignment Framework (STAF)	• Framework for tracing science measurement requirements to project, spacecraft and science instrument requirements • Excel-based	2016 - present	• Demonstrably complete and consistent requirements across all instruments • Improved traceability of engineering requirements to science requirements • Enabled analyses, trades and fault studies to determine science return on different implementation options	(+) Simple but well-conceived tools can enable important conversations between engineers and scientists (+) Process of translating requirements to mathematical constraints helped rqmts validation
Electrical Systems Engineering	• Eclipse/EMF-based authoring tool, git for CM, Leveraging open-source standards (OWL2-DL, SPARQL)	2019-present		

CAESAR Information Integration Platform



openCAESAR: Core elements available as open source software http://www.opencaesar.io/

openCAESAR Enables Rigorous Systems Engineering Practice

Modern systems are becoming more complex than ever before and this complexity will only increase. openCAESAR helps deal with this complexity by providing an advanced platform, on which to define and streamline a rigorous systems engineering methodology.

openCAESAR Has Multiple Value Propositions

Integration of federated information via semantic web vocabularies

Describe the system architecture using tool-neutral semantic vocabulies, model kinds and viewpoints and map them to federated tool-specific counterparts using adapter interfaces.

Continuous automated multi-paradigm analysis via CI/CD workflows

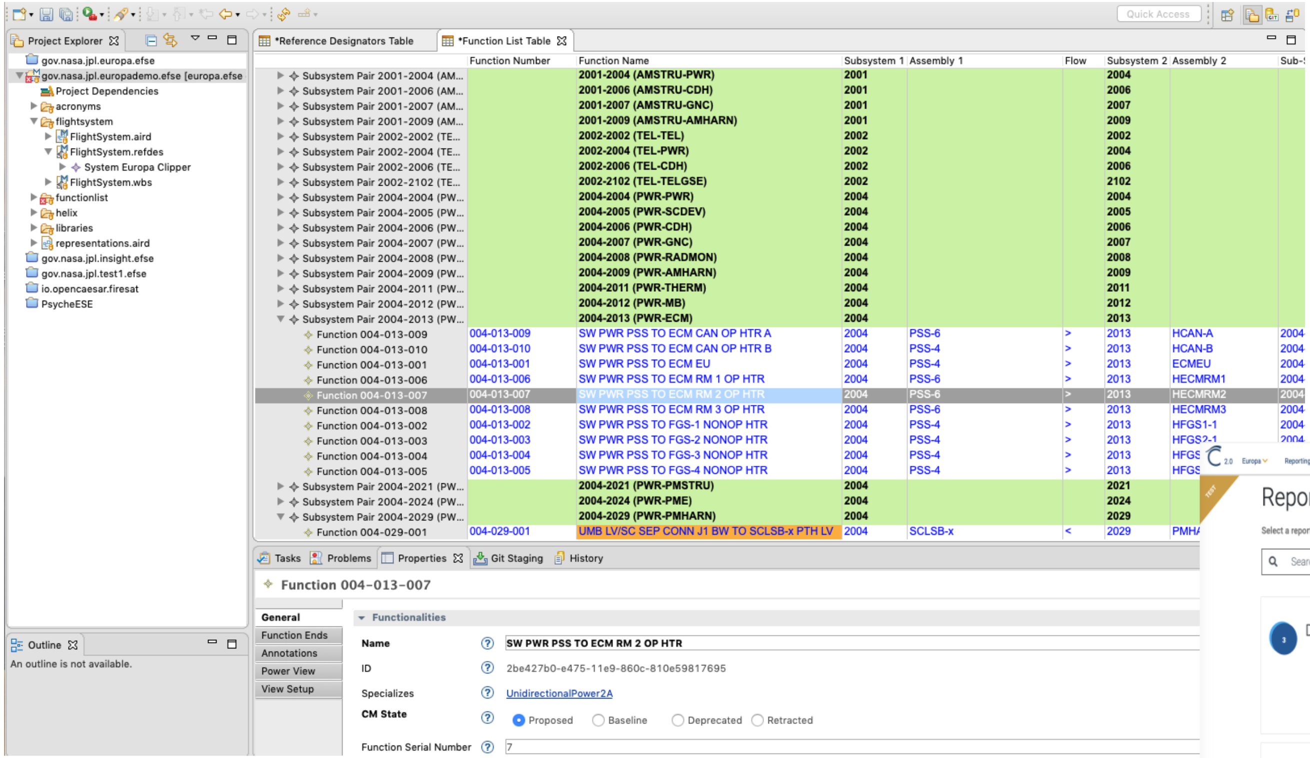
Improve the agility of your systems engineering process by employing DevOps techniques to automate the analysis of your system description using multiple paradigms like ontological analysis and others.

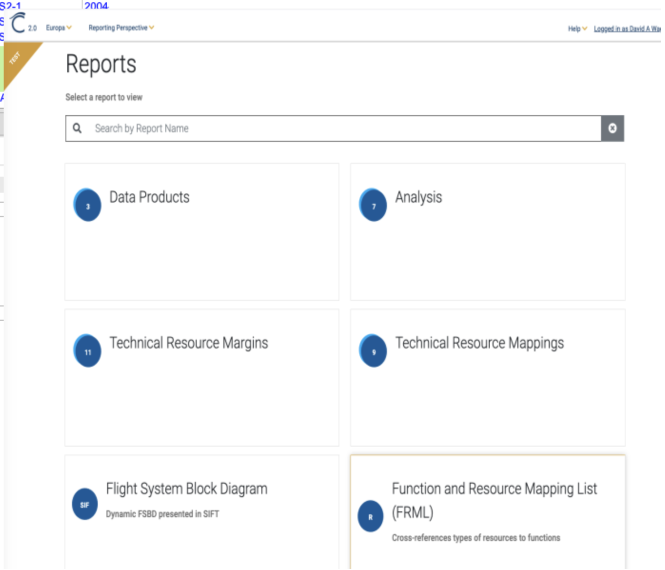
Precise change management via provenance metadata`

Establish a baseline for the integrated system description, manage change proposals using variant configurations, and calculate impact based on the provenance of the changes.

This document has been reviewed and determined not to contain export controlled technical data

Electrical Harness Specification





MBSE Applications on Europa Clipper – with Key Benefits and Lessons

Application	Description	In use	Key Benefits	Key Lessons
Mass Equipment List (MEL)	• SysML/Magid Draw capture and rollup of component mass • Web-based reporting via OpenMBEE	2011-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled early CM	(+) Start small: modest, incremental objectives (+) Produce familiar products with better methods (+) Involve end user continuously (+) Effective project/line collaboration is essential
Power Equipment List (PEL)	• Add Power states/demands to MEL • Provide static description to time-based mission simulation • Web-based reporting via OpenMBEE	2012-present	• Frequent snapshots • Fewer errors • Improved early trades • Enabled integrated simulations	(+) Leverage existing tools (+) Include only the necessary data in the model (-) SysML not ideal for tabular UI (-) Integrate early with Electrical SE model
Power/Energy Simulation	• Multiple tools chained to provide repeatable power demand, energy production,& battery state of charge profiles	2013-present	• Automatic transformation of PEL model resulted in frequent reporting of power and energy resources • Early ability to size system using full mission simulation • Frequent snapshots • Static snapshots in PEL provide validation of time-based simulation	(+) Share time-based profiles with subsystems early on for detailed design (+) Run frequently to detect issues early (+) Validate against high-fidelity subsystem tools (-) Earlier documentation would have helped during leadership change
Architecture & Requirements	• SysML/MD and View Editor (OpenMBEE) -based architecture and requirement development tools	Partial capability 2014; Retired unfinished in 2019	• Architecture Framework provided a useful mental model adapted to JPL missions • Conceptual approach contributed to more complete requirements • Produced reusable stakeholder descriptions	(-) Ambitious developments need risk management (-) Training the full team to think differently is hard (-) Tools and processes need to support end user (-) Parallel developments impede failure analysis …more on slides 10-11
Science Traceability and Alignment Framework (STAF)	• Framework for tracing science measurement requirements to project, spacecraft and science instrument requirements • Excel-based	2016 - present	• Demonstrably complete and consistent requirements across all instruments • Improved traceability of engineering requirements to science requirements • Enabled analyses, trades and fault studies to determine science return on different implementation options	(+) Simple but well-conceived tools can enable important conversations between engineers and scientists (+) Process of translating requirements to mathematical constraints helped rqmts validation
Electrical Systems Engineering	• Eclipse/EMF-based authoring tool, git for CM, Leveraging open-source standards (OWL2-DL, SPARQL)	2019-present	• Correct-by-construction authoring, strong validation • Integration to L4 design w/validation • 95% mission-independent implementation • Reduce manual steps in harness spec & design • Enable SE to specify requirements not design	(+) Continuous integration of SE products is possible (+) Familiar user interfaces lower barrier to entry (+/-) Rigorous approach highlights just how much of current ad-hoc processes need standardizing

What's Next?

- Europa Clipper, in partnership with IMCE, has provided a rich opportunity for innovation and learning.
- Not surprisingly, the results so far serve to remind us that
    - Change is difficult and seldom straightforward.
    - Progress requires patience and steadfast commitment
- We are incorporating the lessons into our approach for the next ten years, with a new set of projects as proving grounds
- We continue to build for the future – laying a solid foundation and exploring new methods that benefit NASA's mission and the aerospace industry as a whole
    - System architecting and design synthesis are two areas which deserve more focus - key to providing innovative & effective solutions
    - JPL is particularly focused on architecture-centric design approaches; completeness, stability and validity of system requirements; and thorough, systematic behavior analysis
    - Increased use of Human-System Interaction (HSI) techniques is helping build more utility and usability into new tools and processes.
    - We are formally adopting a strategy of incrementally building capabilities over multiple project lifecycles - incremental improvements on one project that can be leveraged and grown on the next
    - We are developing an architecture to enable information exchange by engineering teams – based on analysis of user needs & wants
    - We continue to invest in a mix of custom and COTS tooling to support these processes and methods
        - We endeavor to use COTS where possible, to leverage others' investments

- We will continue to push the SE envelope and learn our lessons along the way…

References: Publications on Europa Clipper MBSE

- *An Operations Concept for Integrated Model-Centric Engineering at JPL*, IEEE Aerospace Conference Proceedings, 2010, T. Bayer, L. Cooney, C. Delp, C. Dutenhoffer, R. Gostelow, M. Ingham, J. S. Jenkins, B. Smith.
- *Update – Concept of Operations for Integrated Model-Centric Engineering at JPL*, IEEE Aerospace Conference Proceedings, 2011, T. Bayer, M. Bennett, C. Delp, D. Dvorak, J. S. Jenkins, S. Mandutianu.
- *Model Based Systems Engineering On The Europa Mission Concept Study*, IEEE Aerospace Conference Proceedings, 2012, T. J. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, D. Wagner,
- *Model-Based Systems Engineering Approach to Managing Mass Margin*, in Proceedings of the 5th International Workshop on Systems & Concurrent Engineering for Space Applications (SECESA), Lisbon, Portugal, 2012, Seung H. Chung, Todd J. Bayer, Bjorn Cole, Brian Cooke, Frank Dekens, Christopher Delp, Doris Lam.
- *Early Formulation Model-centric Engineering On Nasa's Europa Mission Concept Study*, Proceedings of 22nd Annual International Symposium (IS2012), Rome, Italy, 2012, T. J. Bayer, S. Chung, B. Cole, B. Cook, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, and D. Wagner.
- *Update on the Model Based Systems Engineering on the Europa Mission Concept Study*, IEEE Aerospace Conference Proceedings, 2013, Todd Bayer, Seung Chung, Bjorn Cole, Brian Cooke, Frank Dekens, Chris Delp, Ivair Gontijo, Dave Wagner.
- *Cloud-based orchestration of a model-based power and data analysis toolchain*, E. Post, K. Dinkel, E. Lee, B. Cole, H. Kim, B. Nairouz, IEEE Aerospace Conference Proceedings, 2016
- *Architecture Modeling on the Europa Project,* G. Dubos , S. Schreiner , D. Wagner , G. Jones , A. Kerzhner , J. Kaderka, AIAA Space Conference Proceedings, 2016
- *A Framework for Writing Measurement Requirements and its Application to the Planned Europa Mission*, S. Susca, L. Jones-Wilson, B. Oaida, IEEE Aerospace Conference Proceedings, 2017
- *A Framework for Extending the Science Traceability Matrix: Application to the Planned Europa Mission*, L. Jones-Wilson, S. Susca, IEEE Aerospace Conference Proceedings, 2017
- *End-to-End Integrated High Fidelity Resource Simulation on Europa Clipper*, Erich Lee, JANNAF 12th Modeling and Simulation Meeting 2018
- *Is MBSE Helping? Measuring Value on Europa Clipper,* T. Bayer, IEEE Aerospace Conference Proceedings, 2018
- *CAESAR Model-Based Approach to Harness Design*, D. Wagner, S.Y. Kim-Castet, A. Jimenez, M. Elaasar, N. Rouquette, S. Jenkins, IEEE Aerospace Conference Proceedings, 2020

For further details about this presentation, see:

- *Europa Clipper: MBSE Proving Ground,* Todd Bayer, John Day, Emma Dodd, Laura Jones-Wilson, Andres Rivera, Narek Shougarian, Sara Susca, David Wagner, IEEE Aerospace Conference Proceedings, 2021

Thanks for listening!

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov

Power/Energy Simulation

Features

- Transforms SysML PEL model to automatically generate a device load model, including available power states and power consumptions of each state
- Uses a detailed component and subsystem schedule with variable resolution to describe transient and steady-state mission scenarios
- Ability to switch between low- and medium-fidelity models for components
- Automated generation of figures and key metrics
- Coupling of high-fidelity tools and mid-fidelity tools for quicker simulations (> 5x speed increase < 1% increase in modeling error)
- Static snapshots in PEL provide validation of time-based simulation
- Easy to manipulate the code to simulate off-nominal conditions
- Ability to run parts of the model independently for quick subsystem or instrument impact assessments

Benefits

- Web interface for results, behaviors and PEL increased team engagement with technical resources
- Enabled quick resolution of trade-studies
- Project-owned power simulation toolchain allows independent validation of subsystem assessments and system level concerns
- Frequent reporting increased visibility into health of the system, as well as triggered early margin recovery exercises
- Potential reusability of modeling components in other projects

Implementation

- Implemented in the Modelica language using Wolfram System Modeler
- Detailed Solar Array shadowing STK model used to generate a lookup table at any given spacecraft vs sun attitude
- Detailed Solar Array radiation damage STK model used to generate time-based lookup table for any mission trajectory
- Ingestion of detailed 5-dimensional (radiation, temperature, flux, current, voltage) hypercube of solar cell data from component testing
- Ingestion of 4-dimensional (temperature, charge, current, voltage) battery cell data from component testing
- Simplified models for replacement heater block and heaters

Challenges

- Maintaining code base consistent with behavior tool, PEL and subsystem descriptions
- Obtaining sufficient subsystem test data as the power system architecture evolved
- Modelica as a language is not widely adopted in space applications. Community is not as large as with other languages such as Simulink, matlab, python, etc.
- Free Modelica tools are not fully compliant with the specification, our model requires a commercial license from Wolfram
- Lack of documentation created difficulty during leadership change

Architecture & Requirements

Features (as envisioned)

- Environment to enable users to collaborate on architecture definition and requirements development – model-centric, as opposed to document-centric
- Database integrating all key elements of architecture description: stakeholders, concerns, scenarios, functions, elements, interfaces, requirements, trades, analyses, models, etc.
- Architecture Framework (AF) adapted/tailored from standards (esp. ISO 42010) and successful prior JPL practice for JPL
- Machine-computable ability to generate text-based requirements from mathematical constraints
- Ability to reconcile requirements taking into account multiple logical decompositions and physical limitations.
- Integration with other IMCE implementations

Implementation

- Early version (Architecture Framework Tool) implemented in Django
- Operational version implementation in MagicDraw (SysML), using View Editor (OpenMBEE) for document generation.
  - used for Project and System-level architecture and requirements only
  - additional custom query/reporting tool developed to get around View Editor search and query limitations
  - content synchronized with DOORS NG repository

Benefits

- Good description of stakeholders and concerns
- Better capture of full requirement rationale in narrative documents
- Automated checking and reporting of requirements characteristics
- Cross-referencing of information to authoritative source

Challenges

- Ambitious approach involved developing methodology, tooling and training in parallel.
- Successful implementation required the entire engineering and management team to be retrained to think in AF terms – this proved to be impractical
- Incomplete tooling and training left users without the means to do the complete architecture definition, including especially requirements
  - Requirements were developed using hybrid of new/traditional approaches. In the end they were late and were found to have significant issues with flowdown and leveling.
  - ViewEditor editing/synchronization capability was not mature or scalable.
- Understanding causes for the failures was difficult due to the highly convolved development
- When a new leadership team came onboard they found the partial implementation too impractical to use and ended up reverting to traditional tools, terminology and methods

Electrical Harness Specification Feature

Features

- Capture system functional composition
  - Assemblies, subsystems, work packages
- Specify abstract electrical interfaces and required interface-to-interface interconnectivity
- Automated validation and generation of web reports and documents
- Integration with TLM/CMD specification data
- Transformation to a form ingestable by ECAD design tool
- Round-trip design compliance validation

Benefits

- Significant improvement in consistency and completeness of specifications as tooling prevents some user mistakes, and reveals others to users
- Users are able to focus on electrical design
- Project-independent vocabulary and analysis promotes reuse and consistent application
- Re-usability at multiple levels: platform, tools, and process
- Familiar user interfaces (e.g., tables) lower barrier to entry

Implementation

- Authoring tool adapts CAESAR workbench by defining vocabulary and discipline views
- Scripted transformations, analysis, reporting automated to run CAESAR engine
- Integrates with Siemens Capital ECAD and other JPL-specific databases
- Continuous integration workflow to merge and validate data and produce downstream products

Challenges

- Discovered many gaps and inconsistencies in traditional process along the way that are now resolved in consistent vocabulary and tooling
- Difficult to eliminate all project-specific concerns
- Strong dependencies on interface data maintained (and CM'd) in documents continues to require manual transcription (e.g., document-based ICDs)

CAESAR Information Integration Platform

Features

- CAESAR defines an architecture and toolkit for federating model data from authoring tools for integration, analysis, and transformation to other forms and reports

Benefits

- Federation approach enables strong configuration and process control not possible if data sources are dynamically “synchronized”
- Formal vocabularies enable meaningful transformation, integration, and cross-analysis of information from multiple sources
- Easier to build model-based tool on top of model-based infrastructure than from scratch

Implementation

- Eclipse/EMF-based authoring tool can be adapted to discipline vocabulary and views (Electrical SE is the first application built with CAESAR)
- Uses git for model data CM (COTS)
- Leverages open-source standards (OWL2-DL, SPARQL) for info representation and access
- Cloud automation and deployment via Kubernetes
- Some parts open sourced at http://opencaesar.io

Challenges

- Getting vocabularies right can be tedious (but you end up having to do this anyway if you need to integrate information)
- Not all tools have interfaces that make information accessible to external automation
- Finding funding to sustain consistent process and tooling at JPL (we can use investment to get started but then what? Projects only want to pay for what they use)